tablon

id: int
name: string
user: Vector<User>
permission: int
visibility: int

targetUsers: Vector<User> messages: Vector<Message>

use: int

get & sets correspondientes
addMessageToTablon(msg);

permission:

0-read permission1-write permission2-read-write permission

visibility:

0-public

1-private (1 user)
2-private (user goup)

"-1" - superPublic (anonymousUser)

targetUsers será:

-null cuando visibility sea 0 ó -1

-vector de tamaño = 1 cuando visibility sea 1

-vector de tamaño > 1 cuando visibility sea 2

use:

-0 valor por defecto significa que el tablón se puede usar de manera presencial -1 significa que el tablón se puede usar de manera tanto presencial como remota. Estamos hablando de los usuarios (targetUsers) ya que el moderador puede verlo de manera remota. (habría que comprobar si pepito.isModerator()==false)

AnonymousUser id: int get y set correpondiente

User

name: string
surname1: string
surname2: string
author: Vector<int>
moderator: Vector<int>

get y sets correspondientes

isAuthor()

int [] isModerator()

setModerator(int idTablon)

int [] isAuthor()
setAutor(int idTablon)

un array de ids de tablon los cuales es autor. setAuthor indica que ese usuario es el autor de dicho tablon

Al igual con el autor, se puede saber que un usuario es moderador de un/os tablon/es isModerator() devuelve un array de enteros que indican el identificador del tablon del cual ese usuario modera. setModerator indica que ese usuario es el moderador de dicho tablon

La figura del dueño del tablón se va a

distinguir, con un método en la clase

user isAuthor() que devuelve

GeneralInfoMessage id: int creator: User message: String image: BufferedImg visibility: int Message soundRecord: Sound... videoRecord:Video... targetUsers: Vector<User> get y sets correspondientes